Selective Audio Filtering for Enabling Acoustic Intelligence in Mobile, Embedded, and Cyber-Physical Systems

Stephen Xia

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2022

© 2022

Stephen Xia

All Rights Reserved

Abstract

Selective Audio Filtering for Enabling Acoustic Intelligence in

Mobile, Embedded, and Cyber-Physical Systems

Stephen Xia

We are seeing a revolution in computing and artificial intelligence; intelligent machines have become ingrained in and improved every aspect of our lives. Despite the increasing number of intelligent devices and breakthroughs in artificial intelligence, we have yet to achieve truly intelligent environments. Audio is one of the most common sensing and actuation modalities used in intelligent devices. In this thesis, we focus on how we can more robustly integrate audio intelligence into a wide array of resource-constrained platforms that enable more intelligent environments. *We present systems and methods for adaptive audio filtering that enables us to more robustly embed acoustic intelligence into a wide range of real-time and resource-constrained mobile, embedded, and cyber-physical systems that are adaptable to a wide range of different applications, environments, and scenarios.*

First, we introduce methods for embedding audio intelligence into wearables, like headsets and helmets, to improve *pedestrian safety* in urban environments by using sound to detect vehicles, localize vehicles, and alert pedestrians well in advance to give them enough time to avoid a collision. We create a segmented architecture and data processing pipeline that partitions computation between embedded front-end platform and the smartphone platform. The embedded front-end hardware platform consists of a microcontroller and commercial-off-the shelf (COTS) components embedded into a headset and samples audio from an array of four MEMS microphones. Our em-

bedded front-end platform computes a series of spatiotemporal features used to localize vehicles: relative delay, relative power, and zero crossing rate. These features are computed in the embedded front-end headset platform and transmitted wirelessly to the smartphone platform because there is not enough bandwidth to transmit more than two channels of raw audio with low latency using standard wireless communication protocols, like Bluetooth Low-Energy. The smartphone platform runs machine learning algorithms to detect vehicles, localize vehicles, and alert pedestrians. To help reduce power consumption, we integrate an application-specific integrated circuit into our embedded front-end platform and create a new localization algorithm called *angle via polygonal regression* (AvPR) that combines the physics of audio waves, the geometry of a microphone array, and a data-driven training and calibration process that enables us to estimate the high resolution direction of the vehicle while being robust to noise resulting from movements in the microphone array as we walk the streets.

Second, we explore the challenges in adapting our platforms for pedestrian safety to more general and noisier scenarios, namely *construction worker safety* sounds of nearby power tools and machinery that are orders of magnitude greater than that of a distant vehicle. We introduce an *adaptive noise filtering architecture* that allows workers to filter out construction tool sounds and reveal low-energy vehicle sounds to better detect them. Our architecture combines the strengths of both the physics of audio waves and data-driven methods to more robustly filter out construction sounds while being able to run on a resource-limited mobile and embedded platform. In our adaptive filtering architecture, we introduce and incorporate a data-driven filtering algorithm, called *probabilistic template matching* (PTM), that leverages pre-trained statistical models of construction tools to perform content-based filtering. We demonstrate improvements that our adaptive filtering architecture brings to our audio-based urban safety wearable in real construction site scenarios and against state-of-art audio filtering algorithms, while having a minimal impact on the power consumption and latency of the overall system. We also explore how these methods can be used to improve audio privacy and remove privacy-sensitive speech from applications that have no need to detect and analyze speech.

Finally, we introduce a *common selective audio filtering platform* that builds upon our adaptive filtering architecture for a wide range of real-time mobile, embedded, and cyber-physical applications. Our architecture can account for a wide range of different sounds, model types, and signal representations by integrating an algorithm we present called *content-informed beamforming* (CIBF). CIBF combines traditional beamforming (spatial filtering using the physics of audio waves) with data-driven machine learning sound detectors and models that developers may already create for their own applications to enhance and filter out specified sounds and noises. Alternatively, developers can also select sounds and models from a library we provide. We demonstrate how our selective filtering architecture can improve the detection of specific target sounds and filter out noises in a wide range of application scenarios. Additionally, through two case studies, we demonstrate how our selective filtering architecture can easily integrate into and improve the performance of real mobile and embedded applications over existing state-of-art solutions, while having minimal impact on latency and power consumption. Ultimately, this selective filtering architecture enables developers and engineers to more easily embed robust audio intelligence into common objects found around us and resource-constrained systems to create more intelligent environments.

Table of Contents

Acknowledgments xii

Dedication xiv

Chapter 1: Introduction 1

1.1 Main Components and Contributions 4

1.1.1 Embedded Acoustic Intelligence for Improving Pedestrian Safety 5

1.1.2 Adaptive Noise Filtering for Improving Construction Worker Safety and Audio Privacy 6

1.1.3 A Common Selective Audio Filtering Framework for Mobile, Embedded, and Cyber-Physical Systems 6

1.2 Contributions to Literature 7

Chapter 2: Embedded Acoustic Intelligence for Improving Pedestrian Safety 9

2.1 Related Work 12

2.2 Studying the Problem 14

2.2.1 Recording Specifications 14

2.2.2 Presence of a Car 16

2.2.3 Direction of a Car 18

2.2.4 Distance of a Car 19

2.3 Overview of PAWS 20

2.3.1 Front-End Hardware 22
2.3.2 Front-End Signal Processing 23
2.3.3 Smartphone Data Processing 25
2.4 PAWS Low-Energy 30
2.4.1 Reducing Power through a Custom Application-Speci c Integrated Circuit . 30
2.4.2 Resolving Constraints Imposed by Utilizing the ASIC 31
2.4.3 Angle via Polygonal Regression 32
2.4.4 PAWS Low-Energy System Architecture 36
2.5 Platform Evaluation 37
2.5.1 Real-Time Performance 37
2.5.2 Power Consumption and Price Breakdown 43
2.6 Real-World Evaluation 44
2.6.1 Experimental Setup 45
2.6.2 Results 47
2.6.3 Limitations and Future Work 51

Chapter 3: Adaptive Acoustic Noise Filtering for Improving Construction Worker Safety 55
3.1 Related Works 57
3.2 Overview of Source Separation and Selective Noise Filtering 58
3.2.1 Spatial Source Separation 58
3.2.2 Content-Based Separation 59
3.3 CSafe Filtering Architecture 61
3.3.1 Content-Based Separation: Probabilistic Template Matching 62

3.3.2 Spatial Separation: A Filter Bank Approach 69

3.3.3 Full Filtering Architecture 72

3.4 CSafe Filtering Architecture Real-World Evaluation 72

3.5 CSafe Platform 79

3.5.1 System Architecture 80

3.5.2 Embedded Hardware Platform 80

3.5.3 Smartphone Platform 81

3.6 System Evaluation 81

3.6.1 Power Consumption 81

3.6.2 Latency 82

3.7 CSafe Platform Real-World Evaluation 83

3.7.1 Vehicle Detection 83

3.7.2 Vehicle Localization 84

3.8 Audio Privacy 85

3.8.1 Related Works 86

3.8.2 PAMS Pipeline 87

3.8.3 Sleeping Event Detection 88

3.8.4 Audio-Based Sleep Monitoring Mobile System Architecture and Design . . 89

3.8.5 Evaluation 89

Chapter 4: A Common Selective Audio Filtering Framework for Mobile, Embedded, and Cyber-Physical Systems 93

4.1 Related Works 95

4.2 Content-Informed Adaptive Beamforming 96

4.3 System 103

4.3.1 Acoustic Localization 103

4.3.2 AvA Architecture 104

4.3.3 Discussion 106

4.3.4 Integrating AvA into New Applications 106

4.4 AvA Evaluation 107

4.5 Case Study: Urban Safety 113

4.5.1 Background 113

4.5.2 Integrating AvA into Acoustic Wearables for Urban Safety 113

4.6 Case Study: Audio Privacy 116

4.6.1 Background 116

4.6.2 Integrating AvA into Mobile Platforms for Sleep Monitoring 118

4.6.3 Improving Command Phrase Privacy in Smart Audio Applications 119

4.7 Discussion 120

4.7.1 Dynamically Changing Sounds to Enhance and Filter 120

4.7.2 Characteristics of Sounds and Models 129

Conclusion 131

References 135

Appendix A: Content-Informed Adaptive Beamforming Model Adaptation and Feature Adaptation 148

A.0.1 Model Adaptation Gradients 148

A.0.2 Feature Adaptation Gradients 150

List of Figures

1.1 Examples of environments and application scenarios that a combined selective l-tering framework could bene t, including: i) home environments: enhancing command phrases while ltering loud music that could make detection more dif cult, ii) urban safety: enhancing vehicle sounds and ltering out construction sounds that make detection more dif cult, iii) audio privacy: for example, ltering out privacy sensitive speech in a mobile sleep monitoring application that records and analyzes only breathing and sleep sounds, iv) entertainment: for example, enhancing the sound of animals at a zoo while ltering wind and other noises, and many more not shown. 2

2.1 An inattentive pedestrian wearing a PAWS headset, and a screen shot of the PAWS application user interface. 10

2.2 Validation system: reference mannequin with eight MEMS microphones and data acquisition board in a low-noise experimental setup. 15

2.3 Spectrogram of one of the recordings from the controlled environment. The car was approaching the mannequin at 25mph. 16

2.4 Distribution of honks and other types of sounds in a 2D feature space. 17

2.5 Normalized relative delays of the microphones for left and right honks. 18

2.6 Relative delay versus time of a car driving past a mannequin from its left to right. . 19

2.7 The maximum cepstral coef cient follows a trend when an approaching car is within about 30m from an observer. 20

2.8 A block diagram of PAWS. The components highlighted in red are portions of the system that are enhanced and modi ed during the second phase of development to create PAWS Low-Energy, which is discussed in Section 2.4. 21

2.9 (Left) Teardown of the PAWS headset; the front-end hardware is exposed inside the left ear housing. (Right) Close up of the PCB that comprises the PAWS front-end hardware. 22

2.10 PAWS Smartphone data processing. The components highlighted in red are portions of the system that are improved upon in PAWS Low-Energy, which is discussed in Section 2.4. 25

2.11 Illustration of the basic idea of non-uniform binning of spectral energy in NBIP, introduced by our collaborators[1]. 26

2.12 (a) The NBIP feature vector for car tire and engine sounds are designed to maximize their dissimilarity from non-car street noises like human chatter, human motions, machine sounds, and loud music. (b) standard MFCC features are not as effective in separating the two classes as our proposed NBIP features. 28

2.13 Plot of relative delay points from playing white noise in eight directions around the user. 35

2.14 AvPR mean and variance interpolation between sampled directions to create the polygon model used to estimate all other directions. 36

2.15 Estimating a new direction with AvPR. Blue dashed lines are example cases of computing the shortest distance from the new point to a segment. The solid black line denotes the path from the new point to the point on the polygon that is closest to the new point. 37

2.16 Pipeline of the PAWS and PAWS Low-Energy feature extraction process. In the PAWS ow, the “Features Calc.” block represents all the operations involved in the features extraction, and “TX” represents the UART communication between the MCU and BLE module. In the PAWS Low-Energy ow, “Extract” refers to the BLE module reading relative delays from the ASIC. 39

2.17 Block diagram of the test setup for the latency between the features from the front-end hardware and the smartphone. 41

2.18 Histogram of PAWS front-end hardware to smartphone latency acquired with the test setup shown in Figure 2.17. 41

2.19 Histogram of PAWS Low-Energy front-end hardware to smartphone latency acquired with the test setup shown in Figure 2.17. We see that the mean latency is shorter than PAWS. 42

2.20 Execution times of various components of the PAWS and PAWS Low-Energy smartphone app. 42

2.21 Experiment scenario in campus street. 45

2.22 Car detection performance. 48

2.23 Metro area car detection confusion matrix. 48

2.24 Estimated direction vs. true direction estimated via AvPR. 49

2.25 Estimated direction vs. true direction estimated via triangulation. 50

2.26 Combined accuracy for PAWS directions prediction. 50

2.27 Car localization accuracy. 51

2.28 confusion matrices for distance estimation. 52

2.29 Estimated distances for cars within 30m are on average 2.8m off of actual distances. 52

3.1 (Left) The CSafe embedded platform, consisting of an an array of four microphones integrated into a feature extraction system introduced in [8]. Two microphones are shown (the front and right microphones), while the other two are on the back and left side of the helmet. (Right) Screenshot of CSafe's smartphone system. 56

3.2 CSafe's full source separation architecture combining PTM (content-based separation) and spatial separation. 61

3.3 (a) Plot demonstrating the extraction process of PTM with a highly con dent "template". We see that if we are more con dent in the frequency component of our sound source (lower variance), PTM will extract coef cients or generated "predicted values" closer to the mean of the model template. b) However, if the model of the frequency component has higher variance (e.g. we are less con dent in value of the frequency component), PTM will extract coef cients further away from the mean of the template. 65

3.4 Examples of ltering results of a vehicle sound mixed over-the-air in the real world with a jackhammer sound. We played the vehicle sound (3.4(a)) and jackhammer sound (3.4(b)) through two different speakers and recorded the mixture with a 4-channel uniform circular microphone array to generate the mixed waveform (3.4(c)). We see from visually inspecting the ltered waveforms that the MM-DenseLSTM neural network approach (3.4(d)) was not able to remove as much of the jackhammer noise as PTM. (3.4(e)). 66

3.5 a) (Left) A wider and less directional lter. (Right) More energy of the original signal is preserved. b) (Left) A more directional lter. (Right) More energy from other directions are ltered out at the cost of greater signal distortion. The signal shown in both scenarios were recorded with a four microphone uniform circular array by playing a vehicle and jackhammer sound with separate speakers on repeat. The ltered signals (right plots) were generated by applying lters (left plots) to diminish the construction tool sound. 71

3.6 Experiment setup at a construction site. 73

3.7 The two plots show a recording of an approaching vehicle in presence of a power tool sound when: a) No noise ltering is applied, and b) CSafe's noise ltering architecture is applied. The green segments highlight the ground truth for when a vehicle is present and the red arrows highlight the segments where the vehicle detector detects the presence of a vehicle. We see in this example, that the vehicle detector can detect the presence of vehicles earlier in each segment after applying our noise ltering architecture. 77

3.8 Average latency of vehicle detection of our vehicle detector without applying noise ltering and after introducing noise ltering. We see that CSafe's noise ltering architecture is able to reduce detection latency from 42.5ms, without any ltering, down to 16ms, which is a greater reduction than any of the other methods tested. . . 79

3.9 CSafe's full system architecture spanning the embedded and smartphone sub-platforms. 79

3.10 Breakdown of latency of each component of CSafe. CSafe completes one cycle of data sampling and computation in 236ms, which is on par with the reaction time of an average person. 82

3.11 PAMS architecture and data pipeline. The lter coef cients, are updated based on the speech templates provided by a Gaussian Mixture noise model and applied in the next window. 88

3.12 PAMS enhanced sleep monitoring system architecture and data pipeline 88

3.13 Confusion matrix metrics for the sleep event detectors of Sleep as Android and PAMS. 90

4.1 Pipeline detailing the CIBF enhancement and ltering optimization process. The forward pass shows the typical steps of evaluating the presence of a sound. The CIBF optimization process is highlighted in red (backward pass), highlighting the three primary steps. Each step is directly tied to one part of the detection pipeline, where connections are highlighted using black dashed arrows. 101

4.2 AvA's System Architecture. 105

4.3 AvA-enhanced urban safety wearable architecture. The embedded hardware platform is borrowed from [8]. AvA directly uses the results from the vehicle detector to determine when to alert the user. 115

4.4 AvA-enhanced platform for ltering out speech and preserving privacy in mobile applications. Unlike in urban safety, audio privacy applications save raw breathing sounds. In the AvA-enhanced systems, we save the ltered signals rather than the raw audio to preserve privacy. 117

4.5 BuMA's system architecture. 123

4.6 Confusion Matrix of multi-class sound detector 124

4.7 Data collection set up. We placed BuMA either to the side or above the mattress, next to where a person would sleep. Then, we played an interfering noise (construction, speech, or music) from a speaker. In this gure, the distance between speaker and the person's head is 150 cm, and BuMA is 30 cm away right above the person lying down on the mattress in the supine position. 126

4.8 Accuracy vs. distance BuMA is away from the head of the person, when BuMA is above the mattress (top) and to the side of the mattress (bottom). The average SNR is listed above each distance. 127

List of Tables

2.1 Power Consumption and Price Breakdown of PAWS and PAWS Low-Energy (PAWS Low-Energy Total in Bold) 43

2.2 Summary of Deployment Events. 46

3.1 Confusion matrix metrics of vehicle detectors under different ltering schemes in a construction site environment. 75

3.2 True positive (recall) for vehicle detection broken down by SNR and tool type of the environment. 75

3.3 Confusion matrix metrics comparing both modes of CSafe with a state-of-art pedestrian safety system [8] 84

3.4 Localization error comparison between CSafe and a state-of-art pedestrian safety system, PAWS [8] 84

3.5 Speech recognition accuracy of recorded clips from PAMS and Sleep as Android [102]. This table lists the proportion of words correctly identi ed in the recorded clips, the proportion of words incorrectly identi ed, and the proportion of words that were not even detected by Google Speech to Text [105]. 90

4.1 Summary of evaluation scenarios and different con gurations of AvA. 107

4.2 Target and noise detection performance in scenario 1 (target: crying and sobbing + noise: construction). 109

4.3 Target and noise detection performance in scenario 2 (target: dog + noise: vehicles). 109

4.4 Target and noise detection performance in scenario 3 (target: piano + noise: speech-noise). 110

4.5 Target and noise detection performance in scenario 4 (target: wild animals + noise: wind). 110

4.6 Performance metrics of vehicle detection of AvA compared to other state-of-art acoustic-based urban safety wearables, while user is having a phone conversation. . 115

4.7 Localization error comparison between AvA and other state-of-art acoustic-based urban safety wearables, while user is having a phone conversation. 115

4.8 Power consumption and latency comparison between an AvA-enhanced wearable and other state-of-art urban safety wearables. The total latency is the time it takes for each system to process one window of audio. The power consumption shows the current draw from each embedded platform powered by a 3.3V battery. 116

4.9 Proportion of words correctly identi ed, incorrectly identi ed, and undetected by Google Speech-to-Text. A lower rate of correctly identi ed words correlates to a more privacy-aware system. 118

4.10 Performance metrics for sleep breathing detection. 119

4.11 Speech intelligibility of the AvA enhanced command phrase mobile application in scenarios where a command phrase was present and not present. 120

Acknowledgements

During my time at Columbia University, I have been extremely fortunate to have support from a cast of amazing and wonderful individuals. I would rst like to express sincere gratitude and thank my advisor, Professor Xiaofan (Fred) Jiang, who has worked tirelessly to enable my fruitful journey. All my work and everything that I have learned along the way is all because of his insight, wisdom, and guidance. Professor Jiang has not only nurtured me as a researcher, but has also contributed to my growth and maturity as a decent human being.

I am also grateful to have opportunities to work with Professor Peter Kinget and Professor Shahriar Nirjon, who have provided their great insight as researchers from other elds and perspectives. I would also like to thank my amazing collaborators, outside of Columbia, who I have shared many conversation and discussions about research and life: Bashima Islam, Md Tamzeed Islam and Shibo Zhang. I would also like give thanks to my former and current colleagues, who have been supporting my research through thick and thin: Rishikanth Chandrasekaran, Daniel de Godoy Peixoto, Jingping Nie, Yanchen Liu, Minghui (Scott) Zhao, Kaiyuan Hou, Asmita Goyanka, Laixi Shi, Ao Liang, Chenye Yang, Hengjiu Kang, Ji Jia, Hugo Meyer, Avik Dhupar, Yigong Hu, Yan Lu, Xiaoqi (Danny) Chen, Jordan Misael Vega, and Andrew Sonta. I would also like to extend my thanks to Professor Romit Roy Choudhury and Professor James Anderson, who have participated on my defense committee.

Along the way, I have met and made many wonderful and amazing friends who have given immense support on my journey: Lei Lei, Mark Barbet, Joe Lee, Ankur Nipane, Anjaly Rajendran, Abhinandan Borah, Han Wang, Shaoru Chen, Jane Zhang, Tingjun Chen, Qinghui Meng, Yian Yu,

Soonhee Moon, Tetsuhiro Harimoto, Tiffany Chien, and Angela Hu. I am extremely fortunate and grateful to have met Peter Wei, who has walked down the same road as me; both of us started down our path towards a Ph.D. the same time and have shared both the ups and downs during our time here. I am also extremely thankful for Tingkai Liu and Zichao (Jack) Wang, who I met as undergraduates, and have inspired me to become a researcher.

Finally, I would like to thank my family and my parents, who have shown unconditional love and support throughout this journey and throughout my entire life.

Financial Support: The research in this thesis was partially supported by the National Science Foundation under Grant Numbers CNS-1704899, CNS-1815274, CNS-11943396, and CNS-1837022. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the of cial policies or endorsements, either expressed or implied, of Columbia University, NSF, or the U.S. Government or any of its agencies.

To my mother and father, Yili Wang and Yunqing Xia.

Chapter 1: Introduction

We are seeing a revolution in computing and arti cial intelligence; intelligent machines have become ingrained in and improved every aspect of our lives. Despite the increasing number of intelligent devices and breakthroughs in arti cial intelligence, we have yet to achieve truly intelligent environments. Many of our cities, homes, and of ces are becoming “smarter" primarily because there is an ever-growing number of smart devices tailored to speci c tasks being released every year. However, these devices are still not being utilized jointly to their maximum potential [1]. There are a wide range of different sensing modalities used in intelligent devices including, but not limited to vision, touch, temperature, and light. Audio is one of the most common sensing and actuation modalities used in intelligent devices. In this thesis, we focus on how we can more robustly integrate audio intelligence into a wide array of resource-constrained platforms that enable more intelligent environments.

We explore methods to embed audio intelligence into wearables, sensor networks, and common objects found all around us to enable a wide range of intelligent audio-based applications. The number of smart devices in our environments is growing rapidly. In the home space, the market for home-based smart devices is expected to grow from USD 102 billion in 2021 to USD 182 billion by 2025 [2]. We are also expecting to see a similar trend in the wearables market, which is projected to grow from USD 116.2 billion in 2021 to USD 265.4 billion by 2026 [3]. Many of these devices use audio as a means to communicate with people, as well as sense and understand the physical world. In 2020, the smart personal audio device grew by 20% [4]. From 2019 to 2025, the smart speaker market, alone, is expected to grow from USD 7.1 billion to USD 15.6 billion [5]. This drastic increase in smart devices and audio devices opens many opportunities, as well as challenges, to make our environments more intelligent and improve our quality of life.

Audio-based smart devices use microphones to listen to the environment and analyze sounds

Figure 1.1: Examples of environments and application scenarios that a combined selective ltering framework could bene t, including: i) home environments: enhancing command phrases while ltering loud music that could make detection more dif cult, ii) urban safety: enhancing vehicle sounds and ltering out construction sounds that make detection more dif cult, iii) audio privacy: for example, ltering out privacy sensitive speech in a mobile sleep monitoring application that records and analyzes only breathing and sleep sounds, iv) entertainment: for example, enhancing the sound of animals at a zoo while ltering wind and other noises, and many more not shown.

using machine learning and signal processing algorithms to execute tasks and provide services.

Often, the system needs to detect and analyze speci c sound(s), while disregarding all others. This

raises two major concerns in embedding acoustic intelligence into resource-constrained mobile, embedded, and cyber-physical platforms. First, there are many instances where the target sound is dif cult to detect and analyze because other noises in the environment can overpower the target sound. Second, many intelligent audio devices listen to the environment for long periods of time, where most of the time the target sound is not present. In these situations, voice devices may be recording and listening to sounds in the environment that may pose privacy risks.

A combined ltering framework, where users can select speci c sounds to keep and enhance or remove and lter may be extremely bene cial to a wide range of different applications. Examples of such applications are shown in Figure 1.1, which is by no means an exhaustive list. For example, in the home environment space, a smart assistant may have trouble detecting spoken commands if loud music or other noises are present. A lter that can enhance speci c command phrases, while ltering out music, would be greatly enhance the usability of the home assistant. In the urban safety space, a selective ltering architecture would enable audio-based safety wearables to lter out loud construction noises that overpower vehicle sounds to better detect and localize oncoming vehicles. In the audio privacy space, a selective ltering architecture that can lter out speech, while retaining or enhancing other sounds in the environment (e.g., breathing sounds for sleep monitoring), would greatly improve the privacy of individuals. As a nal example, in the entertainment space, a selective ltering architecture that can lter out noisy wind and enhance animal sounds in a zoo-based augmented or virtual reality wearable could greatly improve the user experience. There are an in nite number of scenarios where structured noises adversely affect the performance of acoustic detectors or pose a privacy risk, both of which are major challenges in realizing robust acoustic intelligence in many mobile, embedded, and cyber-physical applications.

In addition to challenges related to the detection performance of acoustic-based smart applications, there are a number of systems-based considerations that make this problem even more challenging. First, many smart-devices that are embedded into our homes, of ces, and environments have limited computational resources. Second, many of these applications need to run in real-time. Because of these constraints, the methods we design need to be capable of running on

resource-constrained platforms like mobile phones and embedded microcontrollers; relying on offloading computation to GPU-equipped servers or the cloud is not feasible in many scenarios due to latency.

We present systems and methods for adaptive audio filtering that enables us to more robustly embed acoustic intelligence into a wide range of real-time and resource-constrained mobile, embedded, and cyber-physical systems that are adaptable to a wide range of different applications, environments, and scenarios.

1.1 Main Components and Contributions

In this section, we present an overview of the main contributions of this thesis. We divide the contributions of this thesis into three main components. First, we introduce methods for embedding audio intelligence into wearables, like headsets and helmets, to improve pedestrian safety in urban environments by using sound to detect vehicles, localize vehicles, and alert pedestrians well in advance to give them enough time to avoid a collision. Second, we explore the challenges in adapting our platforms for pedestrian safety to more general and noisier scenarios, namely construction worker safety, where there are sounds of power tools and machinery that are orders of magnitude greater than that of a typical vehicle. We introduce an adaptive noise filtering architecture that allows workers to filter out construction tool sounds and reveal low-energy vehicle sounds to better detect them. We also explore how these methods can be used to improve audio privacy and remove privacy-sensitive speech from applications that have no need to detect and analyze speech. Finally, we introduce a common selective audio filtering framework that builds upon our adaptive filtering architecture for a wide range of real-time mobile, embedded, and cyber-physical applications. This architecture enables developers to select, not only noises to filter out, but also sounds to keep and enhance for their unique applications. This architecture is also capable of supporting a wide range of different detector and sound model types, allowing us to more robustly integrate audio intelligence into a wide range of different mobile, embedded, and cyber-physical applications.

1.1.1 Embedded Acoustic Intelligence for Improving Pedestrian Safety

With the prevalence of smartphones, pedestrians and joggers today often walk or run while listening to music. Since they are deprived of their auditory senses that would have provided important cues to dangers, they are at a much greater risk of being hit by cars or other vehicles. In Chapter 2, we introduce an audio wearable for detecting vehicles, localizing vehicles, and alerting pedestrians in advance to prevent pedestrian-vehicle accidents. We embed a headset wearable with an array of microphones and an embedded front-end platform. We create a segmented architecture and data processing pipeline that partitions computation between embedded front-end platform and the smartphone platform. The embedded front-end hardware platform consists of a microcontroller and commercial-off-the shelf (COTS) components and samples audio from an array of four MEMS microphones. Our embedded front-end platform computes a series of spatiotemporal features used to localize vehicles: relative delay, relative power, and zero crossing rate. These features are computed in the embedded front-end headset platform and transmitted wirelessly to the smartphone platform because there is not enough bandwidth to transmit more than two channels of raw audio with low latency using standard wireless communication protocols, like Bluetooth Low-Energy. Finally, the smartphone platform runs machine learning algorithms to detect vehicles, localize vehicles, and alert pedestrians.

To help reduce power consumption, we integrate an application-speci c integrated circuit (ASIC effort led by Daniel de Godoy [6]) into our embedded front-end platform in the second phase of this work. However, because the ASIC computes only one speci c feature that we use in localizing vehicles, we lose many of the spatio-temporal features we required to estimate the location of vehicles. As such, we create a new localization algorithm called angle via polygonal regression (AvPR) that combines the physics of audio waves, the geometry of a microphone array, and a data-driven training and calibration process that enables us to estimate the high resolution direction of the vehicle while being robust to noise resulting from movements in the microphone array as we walk the streets.

1.1.2 Adaptive Noise Filtering for Improving Construction Worker Safety and Audio Privacy

In Chapter 3, we adapt our pedestrian safety platform, introduced in Chapter 2, for improving construction worker safety. Because construction workers are using power tools and machinery that are orders of magnitude greater than that of oncoming vehicles, our audio wearable pedestrian safety platform had dif culty hearing and detecting vehicles. To address this challenge, we introduce an adaptive audio noise ltering architecture to lter out construction sounds, enabling us to better detect vehicle sounds. Our architecture combines the strengths of both the physics of audio waves and data-driven methods to more robustly lter out construction sounds while being able to run on a resource-limited mobile and embedded platform. In our adaptive ltering architecture, we introduce and incorporate a data-driven ltering algorithm, called probabilistic template matching (PTM), that leverages pre-trained statistical models of construction tools to perform content-based ltering. We demonstrate improvements that our adaptive ltering architecture brings to our audio-based urban safety wearable in real construction site scenarios and against state-of-art audio ltering algorithms, while having a minimal impact on the power consumption and latency of the overall system.

We also explore how our adaptive ltering architecture can be used to improve the privacy of smart home applications that use audio to analyze speci c sounds in the environment (e.g., breathing sounds for monitoring sleep quality), but can also record other non-required privacy-sensitive signals (e.g., speech). In these applications, we show how our adaptive noise ltering architecture can be used to remove privacy-sensitive signals, like speech, while minimally impacting the performance of detecting target sounds.

1.1.3 A Common Selective Audio Filtering Framework for Mobile, Embedded, and Cyber-Physical Systems

In Chapter 4 we extended the adaptive ltering architecture introduced in Chapter 3 to a wide range of different applications and scenarios. Speci cally, Chapter 3 gave us insight into how audio ltering could be used to improve the detection performance of vehicles by ltering out

construction sounds in a construction site, where we expect both vehicles and construction noises to be present. From this work, we realize that there are numerous applications, beyond urban safety, that could potentially bene t from an audio ltering architecture that enables developers and engineers to select speci c sounds to enhance and keep (target sounds) while ltering out and removing other sounds they expect to interfere with their systems (noises). We accomplish this goal by introducing a common selective audio ltering framework that is adaptable to a wide range of different sounds, noises, and applications. Our architecture can account for a wide range of different sounds, model types, and signal representations by integrating an algorithm we create called content-informed beamforming (CIBF). CIBF combines traditional beamforming (spatial ltering using the physics of audio waves) with data-driven machine learning sound detectors and models that developers may have already created for their own applications to enhance and lter out speci ed sounds and noises. Alternatively, developers can also select sounds and models from a library we provide.

We demonstrate how our selective ltering architecture can improve the detection of speci c target sounds and lter out noises in a wide range of application scenarios. Additionally, through two case studies, we demonstrate how our selective ltering architecture can easily integrate into and improve the performance of real mobile and embedded applications over existing state-of-art solutions, while having minimal impact on latency and power consumption.

1.2 Contributions to Literature

The audio wearable platform for improving pedestrian safety was published in the ACM/IEEE International Conference on Internet of Things Design and Implementation 2018 (IoTDI'18) [7] and the IEEE Internet of Things Journal 2019 (IoTJ'19) [8]. This research also received the Best Demo Runner Up Award (ACM SenSys 2016) [9], Best Demo Award (ACM/IEEE IoTDI'18) [10], Best Presentation Award (IEEE VNC'18), and Best App Runner Up Award (IEEE VNC'18) [11]. This work has also been featured by various news and media outlets including New York Post, IEEE Spectrum, Fast Company, Mashable, Gizmodo, The Telegraph, Engineering.com, India

Times, and IEEE Signal Processing Magazine. Our work on adaptive audio noise ltering for construction worker safety and audio privacy has been published in the ACM/IEEE International Conference on Information Processing in Sensor Networks 2021 (IPSN'21) [12] and the Second International Workshop on Challenges in Arti cial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT'20) [13]. The common selective audio ltering platform for mobile, embedded, and cyber-physical systems will appear later this year in the ACM/IEEE International Conference on Information Processing in Sensor Networks 2022 (IPSN'22).

Chapter 2: Embedded Acoustic Intelligence for Improving Pedestrian Safety

Smartphones have transformed our lifestyles dramatically, mostly for the better. Unfortunately, smartphone usage while walking has become a serious safety problem for many people in urban areas around the world. Pedestrians listening to music, texting, talking or otherwise absorbed in their phones are putting themselves at risk by tuning out the traf c around them [14], as reported by the Washington Post. Since a pedestrian is deprived of auditory input that would have provided important cues to dangers such as honks or noises from approaching cars, he or she is at a much greater risk of being involved in a traf c accident. We have seen a sharp increase in injuries and deaths from such incidents in recent years. According to a study by Injury Prevention and CNN, the number of serious injuries and deaths occurring to pedestrians who were walking with headphones has tripled in the last seven years in the United States [15]. This phenomenon affects cities globally, and is an important societal problem that we want to address by introducing advanced sensing techniques and intelligent wearable systems.

In this chapter, we present PAWS, Pedestrian Audio Wearable System, aimed for urban safety. The PAWS project is in collaboration with my colleagues both at Columbia University and the University of North Carolina at Chapel Hill. PAWS is a low-cost headset-based wearable platform that combines four MEMS microphones, signal processing and feature extraction electronics, and machine learning classi ers running on a smartphone to detect and locate imminent dangers, such as approaching cars, and warn pedestrians in real-time. Figure 2.1 shows PAWS in action.

We take an audio approach to detect, localize, and alert users to oncoming vehicles in real-time and in a resource-limited wearable. Works that leverage sensors that provide more rich information about the environment, such as camera or LIDAR, have high computational and battery requirements, making it dif cult to create a long-lasting battery-powered platform. Although audio does not provide as rich information about the environment as sensors like cameras, standard micro-

Figure 2.1: An inattentive pedestrian wearing a PAWS headset, and a screen shot of the PAWS application user interface.

phones provide 360 degree coverage and require less computation to sample and extract features.

With newer smartphones equipped with multiple built-in microphones, it may be tempting to re-purpose those microphones in software to localize cars based on common localization techniques. However, these approaches require the user to constantly hold their phones steady and to not block the built-in microphone while walking [16][17]. Further, most built-in microphones are designed for voice and are often band-limited. These two limitations prevent the smartphone from capturing useful features produced by approaching cars in realistic urban environments.

This is a challenging problem as the battery-powered wearable platform needs to detect, identify, and localize approaching cars in real-time, process and compute large amounts of data in an energy and resource constrained system, and produce accurate results with minimal false positives and false negatives. For example, if a user's reaction-time is 500 ms, the system has 360 ms to detect a 25 mph car and alert the user when it is 16 m away. This problem is further compounded by high levels of mixed noise, typical of realistic street conditions.

We address these challenges over two phases of design and development. In the first phase (PAWS), we develop a segmented architecture and data processing pipeline that partitions com-

putation into processing modules across a front-end hardware platform and a smartphone. The microcontroller-based front-end hardware platform consists of commercial off-the-shelf (COTS) components embedded into a standard headset and collects four channels of audio from four MEMS microphones that are strategically positioned on the headset. Temporal-spatial features such as relative delay, relative power, and zero-crossing rate are computed inside the front-end platform using the four channels and transmitted wirelessly to a smartphone. A fth standard headset microphone is also connected to the audio input of the smartphone, and together with the data sent from the front-end platform, classi ers are trained and used to detect an approaching car and estimate its azimuth and distance from the user.

In the second phase of development (PAWS Low-Energy), we tackle the challenge of power consumption through the design and implementation of an application-speci c integrated circuit to extract some of the computationally expensive features. We also develop new methods to increase the accuracy and granularity of our audio-based vehicle localization. We evaluate PAWS using both controlled experiments inside parking lots and real-world deployments on urban streets.

We summarize the major points of this chapter below:

- We incorporate a new acoustic feature, Non-Uniform Binned Integral Periodogram, which is designed to capture frequency domain characteristics of low-frequency noise-like sounds, such as the sound produced by the friction between a car's tires and the road. We develop classi ers to recognize and localize cars approaching the user in real-time.

- We create PAWS, a low-cost, end-to-end wearable system using COTS components accompanied with a smartphone application to provide real-time alerts of oncoming cars to pedestrians in noisy urban environments. We demonstrate that inattentive pedestrians can immediately bene t from our system.

- We present a second system, PAWS Low-Energy, that improves upon the power consumption of our COTS implementation by of oading critical and computationally expensive features onto an application-speci c integrated circuit, developed by our collaborators [6]. We

additionally introduce Angle via Polygonal Regression (AvPR), an easily calibrated method for estimating the direction of arrival of car sounds. AvPR improves upon the granularity of direction estimates over the classi cation approach employed in PAWS and accommodates for noise better than classical geometric approaches (e.g. triangulation) for estimating direction, while remaining computationally inexpensive.

- We develop a segmented architecture and data processing pipeline that intelligently partitions tasks across the front-end hardware and the smartphone and ensures accuracy while minimizing latency.

As the industry is investing heavily in intelligent headphones [18][19], our hardware-software co-design approach presents a compelling solution towards protecting distracted pedestrians in urban environments.

2.1 Related Work

Object recognition and localization have been vastly explored in the literature. Almost all of them mirror techniques that are present in nature, such as the use of stereo imaging [20], ultrasonic radars [21], and acoustic source localization [22]. In vehicular tracking, video based approaches have been widely used [23][20][24]. The amount of information that can be extracted from images is undoubtedly greater than any other types of sensors. Commonality in vehicles' shapes and standardized road signs have enabled machine learning algorithms to identify and predict the movement of cars [25]. Although such systems offer outstanding solutions for devices that can be hosted in large platforms, e.g. in an autonomous car for collision prevention [26], these are not suitable for use in wearable systems. A major limitation is the high computational requirements for real-time image processing. Another major issue is the privacy of the user. Video can reveal an alarming amount of personal identi able information.

Active techniques like radar and LIDAR can certainly be used to detect the presence of obstacles and even some of its spatial behaviors [27][28], but such solutions face great challenges

in classifying what those obstacles are. This is particularly problematic in urban environments where moving and stationary obstacles are abundant, but only a few are real threats to the user. On the implementation side, the inherently high power dissipation of active transducers are usually discouraging for portable devices.

Passive audio sensors, on the other hand, provide enough information to allow classi cation and localization of the source with less computational and power requirements. But, unlike other techniques already published [29][22][30], PAWS uses machine learning algorithms to improve its predictions. By doing so, the system requires a large amount of learning data, but gains in exibility, speed, and complexity. Audio classi cation has been used for event detection like, coughing detection [31], gun shot detection [32], human activity (e.g. talking, crying, running etc) detection [33]. These works mostly focus on identifying prominent sounds like gun shots or shouting rather than noise-like car sounds. [34] classi ed subtle sounds like keyboard typing, door knock etc. but all of the sounds were in an isolated environment not in real life noisy environment. [35] considered bus and trucks as events but had a very low accuracy of 24%. Other signals like video [36] [37] or seismic signals [38] have been used for vehicle detection but are not suitable for a wearable system like PAWS.

Other works that leverage sensors to enhance pedestrian safety utilize sensors placed on the shoe [39] or the camera on the smartphone [40], but these approaches are either unable to detect and localize cars or provide limited coverage.

In recent years, developments in vehicle to vehicle, pedestrian, and infrastructure communications are beginning to allow cars and pedestrians to directly communicate with each other in close proximity. Dedicated Short Range Communication (DSRC), a wireless protocol developed specifically for vehicular networking, is becoming a popular protocol in vehicular networks to transmit information and alerts [41][42], but is not natively supported on smartphones and require modi - cations to existing vehicles. Solutions that leverage standard WiFi or cellular protocols generally require every car to have a wireless transmitter [43][44], do not meet timing and latency requirements [45][46], or modify the SSID of smartphone WiFi beacons [47] and cannot be implemented

on a smartphone with standard privileges. Additionally, a single pedestrian only requires our custom headset to receive the full functionality of PAWS and PAWS Low-Energy. In contrast, V2X systems commonly require modi cations to support speci c wireless protocols on all passing cars before a single pedestrian can bene t.

2.2 Studying the Problem

Before developing PAWS into a wearable system, we studied the car sound recognition and localization problem using a validation platform. The objective of this exercise was to analyze the feasibility and complexity of our proposed solution and to determine the speci cations required to capture the necessary information, e.g., audio sampling rate, sensor placement, and most relevant features to use in the machine learning algorithms.

As shown in Figure 2.2, the platform directly connects eight MEMS microphones to a computer. The microphones were placed on a mannequin head to reproduce the physical phenomena of the nal setup, such as the acoustic shadow of the human head [48] and the approximate spacing among sensors on a real user.

The study has been done in ve different locations in two different cites: a metropolitan area and a college town. The locations were two parking spaces, a four-way intersection, and two multi-lane streets. We analyzed recorded audio from 47 different cars. Other than the parking spaces, where we conducted our rst set of controlled experiments with labeled distances, directions, and precise time-keeping of honks and car passing, all other scenarios were uncontrolled.

2.2.1 Recording Speci cations

In order to characterize the sounds of interest, such as an approaching vehicle's tire friction, engine noise, and honks, we conducted controlled experiments in two parking lots (Figure 2.2 shows one of the experiments). These results are later cross-checked against uncontrolled experiments' for consistency.

Figure 2.3 shows the spectrogram of one of the recordings from the controlled experiments.

Figure 2.2: Validation system: reference mannequin with eight MEMS microphones and data acquisition board in a low-noise experimental setup.

Both the top and the bottom gures correspond to the same recording. Approximately 5s after the recording starts, a car honks, resulting in distinct stationary tones with fundamental frequencies near 500Hz. The vehicle then accelerates towards the mannequin. In the bottom gure, where the lower part of the spectrogram is highlighted, we see the engine noise. The engine noise follows its RPM. In an automatic car, the engine noise is bounded between 60Hz and 200Hz (at the 7 seconds mark, the shift in the engine gear is noticeable). Once the vehicle gets closer to the mannequin, the friction noise from the tires and asphalt gets louder. This noise has a band-limited spectrum with more energy below 3kHz. When the car crosses the system near the 12s mark, a burst of air causes a loud white noise. Similar spectrum components were found on several recordings of different cars at similar speeds (20-30mph) on dry asphalt.

These observations indicate that to identify warning honks and vehicles that are still approaching the user, the system audio must reliably capture frequencies from 50Hz to 6kHz. This requirement means that the system needs custom microphone drivers with a cut-off frequency of less than 10Hz (in contrast to standard headset microphones with approximately 100Hz cut-off frequency) and analog-to-digital converters with sampling rates above 12kSamples/s.

Figure 2.3: Spectrogram of one of the recordings from the controlled environment. The car was approaching the mannequin at 25mph.

2.2.2 Presence of a Car

The presence of a car can be determined from high-energy, sharp sounds like honks, as well as from low-energy, noise-like sounds such as the sound of friction between a tire and the road. Being able to detect cars based on friction noise is crucial given the increasing popularity of electric cars with quieter engines.

Honks are louder and, thus, easier to detect than car tire or engine sounds. We analyze the Mel-Frequency Cepstral Coef cients (MFCC) [49] of honks and compare them with non-honk street sounds. We start with MFCC, since it is one of the most commonly used acoustic features for detecting various types of sounds [50][51][52][53] including car sounds [54]. For visualization

Figure 2.4: Distribution of honks and other types of sounds in a 2D feature space.

purpose, we reduce the 13-dimension MFCC features to two dimensions (using PCA [55]) and the result is shown in Figure 2.4. We observe that honks are separable from other sounds as they cluster around a different point in space. Honks are easily detectable using all 13 coef cients.

MFCCs, however, are not effective in detecting other types of car noises such friction between tires and the road. The fundamental reason behind this is that the Mel-scale, expressed by $2595 \log_{10}(1 + f/700)$, was originally designed to mimic human hearing of speech signals that maps frequencies < 1kHz somewhat linearly, and maps > 1kHz logarithmically. Our analysis on tire friction sounds shows that about 60% signal energy is attributed to frequency components below 1 kHz. Hence, to model such low-energy, low-frequency, noise-like sounds, we need to develop a new feature that captures these sub-kHz characteristics. We propose this new feature in Section 2.3.3.

Figure 2.5: Normalized relative delays of the microphones for left and right honks.

2.2.3 Direction of a Car

To determine the direction, we record audio of cars approaching from different directions and analyze their effect on the microphone set. Some of these recordings also have honks in them. Intuitively, microphones that are closer to the sound source and are not obstructed by the human head should receive signals earlier, and the signals should be stronger. Hence, the relative delays and the relative energy of the received signals should be strong indicators of the direction of an approaching car.

In Figure 2.5, we plot the relative delays of the microphones with respect to the front microphone for left and right side honks. We see that the relative delays change signs for left and right honks. We do similar tests with eight directions (each covering a $\approx 22.5^\circ$ cone surrounding the mannequin) to successfully determine the directions of honks near the user.

Similarly, we plot the relative delays of the microphones for a car that passes the mannequin from its left to the right (Figure 2.6). We observe that the relative delays are quite random on both left and right ends. As the car approaches the mannequin, we see a trend in all the curves with one or more of them reaching their peaks. The trend reverses as the car passes the mannequin. This behavior suggest that patterns in relative delays (when they are looked at together) are useful to determine the direction of passing. Hence, by learning the trend and the point when the trend

Figure 2.6: Relative delay versus time of a car driving past a mannequin from its left to right.

reverses, it is possible to differentiate between a car on the left from a car on the right, as well as their angular directions.

2.2.4 Distance of a Car

In an attempt to estimate the distance, we formulate a regression problem that maps sound energy to distances. Later we realize that due to environmental noise and the weakness of car sounds, a ne grained location estimation is extremely inaccurate when the car is farther than 30m from the audio recorder. When the car is within 30m, we nd that the maximum value of the cepstral coef cients (computed every 100 ms) is approximately linearly correlated with distance, as shown in Figure 2.7 for a car that is driven toward the mannequin. This relationship can be exploited to form a regression problem that maps maximum cepstral coef cients to distances.

For cars farther than 30m, although we are able to detect their presence and estimate their direction, a precise distance estimation results in a large error. However, we learn that the distance estimation problem can be formulated as a multi-class classi cation task by dividing the absolute distances into a number of ranges such as (0, 20m], (20m, 40m], and (40m, 60m]. Each of these ranges can be characterized with additional features, such as zero-crossing rate, and can be classi ed accurately using a machine learning classi er.

As such, one option is to use a two-level approach for distance estimation. The rst level em-

Figure 2.7: The maximum cepstral coef cient follows a trend when an approaching car is within about 30m from an observer.

ploys a classi er to determine a coarse-grained distance range, and if a car is detected within the nearest range, it applies regression to obtain a ne grained distance estimate. A second option is to disregard estimating coarse-grained distance and only provide ne-grained distance measurements via regression. As shown in Figure 2.7, for distances above 30 meters, the maximal cepstral coef cient maps to approximately the same value. As such, the regression model inherently classi es car distances into two coarse groups: greater than 30 meters and less than 30 meters away. We considered both methods over the course of designing both phases of PAWS, as detailed in Sections 2.3 and 2.4.

2.3 Overview of PAWS

PAWS is a wearable headset platform and smartphone application that uses ve microphones and a set of machine learning classi ers to detect, identify, and localize approaching cars in real-time and alerts the user using audio/visual feedback on his smartphone.

The system consists of three main components: sensors and their drivers, front-end hardware for multi-channel audio feature extraction, and a smartphone host for machine-learning based vehicle detection and localization, which are shown in Figure 2.8. Four of the MEMS microphones,

Figure 2.8: A block diagram of PAWS. The components highlighted in red are portions of the system that are enhanced and modi ed during the second phase of development to create PAWS Low-Energy, which is discussed in Section 2.4.

labeled MIC1 to MIC4, are distributed over the user, at the left and right ear, back of the head, and chest of the user, to provide relevant information about the sound source's location. The front-end hardware synchronously acquires analog signals from these microphones and locally extracts acoustic features that are used by a smartphone application. PAWS performs signal processing inside the front-end hardware so that only features need to be transmitted wirelessly to the smartphone (via BLE) instead of large amounts of raw audio data. The front-end hardware is a battery-powered embedded platform that is housed within the con nes of the headset that uses its own set of microphones for sound processing. It does not interact with the speakers or microphone of the headset. As such, a user would have not experience any degradation in sound or microphone quality of the headset.

The standard microphone of the headset (the fth microphone, MIC5) is connected to the 3.5mm audio input of the phone. Data from the fth microphone is directly acquired by the smartphone. The audio from the smartphone/headset microphone is acquired in the same way as com-